

REMARKS/ARGUMENTS

This Amendment is in response to the Office Action of January 10, 2008. Claims 10-28 are pending in the present application. Claims 10-28 have been rejected. Claims 10-15 and 22-28 have been canceled. Accordingly, claims 16-21 remain pending. Also, the Title and Abstract have been amended to reflect the cancellations to the claims. For the reasons set forth more fully below, Applicants respectfully submit that the claims as presented are allowable. Consequently, reconsideration, allowance, and passage to issue are respectfully requested.

Applicants do not concede in this application that the claims as previously presented are not patentable over the art cited by the Examiner, as the present claim amendments and cancellations are only for facilitating expeditious prosecution of the remaining claims. Applicants respectfully reserve the right to pursue these and other claims in one or more continuations and/or divisional patent applications. Nevertheless, Applicant respectfully submits that the pending claims are now in condition for allowance.

Specification

Examiner Stated:

The specification is objected to...the use of “computer-readable medium” is not clearly defined and lacks support from the specification.

As noted above, claims 22-28 (computer-readable medium claims) have been canceled. Accordingly, Applicants respectfully submit that the objection is now moot.

Rejections Under 35 U.S.C. §102

Examiner Stated:

Claims 10-28 are rejected under 35 U.S.C. 102(e) as being taught by Goldberg et al. (U.S. Patent 6,496,833)....

Applicants respectfully disagree with the Examiner's rejections. As noted above, claims 10-15 and 22-28 have been canceled. The present invention provides a method for supporting a plurality of graphical user interface (GUI) application programming interfaces (APIs). In accordance with one embodiment of the present invention, the method includes translating a plurality of elements of a query model into objects that are independent of any type of data structure associated with the plurality of GUI APIs, the plurality of elements being translated through use of a model content provider in communication with the query model, the plurality of elements representing a database statement. The method also includes passing the translated objects from the model content provider to a first content viewer in communication with the model content provider, the first content viewer supporting multiple GUI APIs. The method also includes passing the translated objects from the first content viewer to a second content viewer, the second content viewer being in communication with the first content viewer and an application written to run on a specific GUI API of the plurality of GUI APIs. The method also includes using the second content viewer to manipulate the translated objects into one or more types of data structures required by the specific GUI API for use by the application. Goldberg does not teach or suggest these features, as discussed below.

Goldberg discloses a query object generator tool is used to generate interface definitions and source code which implement a database query object. The tool allows a client to construct a query object without being familiar with the underlying database language and without being concerned with programming details such as concurrency problems and connection management. The tool consists of an internal state object which represents the query object, including information which can be saved to reconstruct the query object at a later date, and code generator objects which generate the code required to implement the query object defined by the internal state object. The code generator objects are arranged in a hierarchy so that a generator object can be instantiated which generator object is specific to the database to be accessed and the language to which the implementation is targeted. An optional graphic user interface (GUI) may also be provided to allow a user to interact with the tool. (Abstract.)

However, Goldberg does not teach or suggest the “translating a plurality of elements of a query model into objects that are independent of any type of data structure associated with the plurality of GUI APIs,” as recited in independent claim 16. The Examiner has referred to Figure 7 elements 702 and 704 of Goldberg as describing the elements of a query model that are translated into objects, and also referred to column 8, lines 20-47, of Goldberg as describing translated objects that are independent of any type of data structure associated with the plurality of GUI APIs. However, element 702 is merely a “QueryInfo class” that “contains methods that translate parameter notations between a notation entered by the user and the notation actually used by the underlying

database,” and element 704 is merely a “Parameter class” that “defines data structures that describe the query input parameters” (column 11, lines 44-60). These classes are not themselves elements that are translated into objects but are instead “part of the query object internal state 602 of query object generator tool 600.”

Furthermore, nowhere does Goldberg even describe how objects that may be associated with these classes are “independent of any type of data structure associated with the plurality of GUI APIs” as in the present invention. The Examiner has asserted that IDL code is not associated with an API, referring to column 8, lines 20-47, of Goldberg. However, this section merely describes a “code snippet,” which references a “ZZZ/ZZZ.idl-IDL interface.” Nowhere does Goldberg specifically teach or suggest that objects, or the IDL code, are independent of any type of data structure associated with the plurality of GUI APIs. In fact, column 7, line 64, to column 8, line 4, of Goldberg states that the “query object generator tool preferably generates a query object for use with a conventional CORBA system” and that the “query generator tool 516 generates a CORBA IDL interface code 518,” which suggests a CORBA-based API, and thus teaches away from “translating a plurality of elements of a query model into objects that are independent of any type of data structure associated with the plurality of GUI APIs,” as in the present invention.

Therefore, Goldberg does not teach or suggest the combination of steps as recited in independent claim 16, and this claim is thus allowable over Goldberg.

Dependent claims

Dependent claims 17-21 depend from independent claim 16. Accordingly, the above-articulated arguments related to independent claim 16 apply with equal force to claims 17-21, which are thus allowable over the cited reference for at least the same reasons as claim 16.

CONCLUSION

Applicants' attorney believes this application is in condition for allowance. Should any unresolved issues remain, Examiner is invited to call Applicants' attorney at the telephone number indicated below.

Respectfully submitted,

April 10, 2008

/Joseph A. Sawyer, Jr./
Joseph A. Sawyer, Jr.
Reg. No. 30,801

Customer Number 45728
(650) 493-4540
(650) 493-4549